



Losst

[Главная](#) ▸ [Инструкции](#) ▸ [Компиляция программ Linux](#)

КОМПИЛЯЦИЯ ПРОГРАММ LINUX

[Инструкции](#)

🕒 Декабрь 10, 2016

💬 8

👤 admin

Все программы для компьютера представляют собой набор команд процессора, которые состоят из определенного набора бит. Этих команд несколько сотен и с помощью них выполняются все действия на вашем компьютере. Но писать программы непосредственно с помощью этих команд сложно. Поэтому были придуманы различные языки программирования, которые проще для восприятия человеку.

Для подготовки программы к выполнению, специальная программа собирает ее из исходного кода на языке программирования в машинный код - команды процессора. Этот процесс называется компиляция. Linux - это свободное программное обеспечение, а поэтому исходные коды программ доступны всем желающим. Если программы нет в репозитории или вы хотите сделать что-то нестандартное, то вы можете выполнить компиляцию программы.

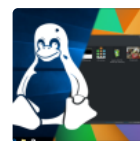
ПОИСК:



Начните изучать
Linux прямо сейчас!

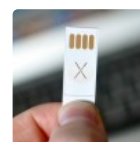
[Карта сайта](#)

Как пользоваться
редактором Vim

[Полезно](#)[Лучшие](#)[Последние](#)[Метки](#)

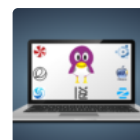
Установка
Linux рядом
с Windows 10

ЯНВ 19, 2016



Как сделать
загрузочную
флешку
Ubuntu

ЯНВ 4, 2016



Лучший
Linux для
ноутбука
2019

МАЙ 8, 2016



программ Linux, как происходит процесс компиляции, а также рассмотрим насколько гибко вы сможете все настроить.

Содержание статьи:

- [Подготовка системы](#)
- [Как выполняется компиляция?](#)
- [Компиляция программ Linux](#)
 - [Получение исходников](#)
 - [Настройка configure](#)
 - [Сборка программы](#)
- [Выводы](#)

ПОДГОТОВКА СИСТЕМЫ

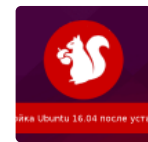
Мы будем компилировать программы, написанные на Си или C++, так как это наиболее используемый язык для программ, которые требуют компиляции. Мы уже немного рассматривали эту тему в статье установка из tar.gz в Linux, но та статья ориентирована больше на новичков, которым нужно не столько разобраться, сколько получить готовую программу.

В этой же статье тема рассмотрена более детально. Как вы понимаете, для превращения исходного кода в команды процессора нужно специальное программное обеспечение. Мы будем использовать компилятор GCC. Для установки его и всех необходимых инструментов в Ubuntu выполните:

```
$ sudo apt install build-essential manpages-dev git automake autoconf
```

```
sergii@sergii-VirtualBox:~$ sudo apt install build-essential manpages-dev
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
build-essential is already the newest version (12.1ubuntu2).
build-essential установлен вручную.
Пакеты, которые будут обновлены:
manpages-dev
обновлено 1, установлено 0 новых пакетов, для удаления отмечено 0 пакетов, и 728 пакетов не обновлено.
Необходимо скачать 2.048 kB архивов.
После данной операции, объем занятого дискового пространства возрастёт на 30,7 kB.
Хотите продолжить? [Д/н] y
Get:1 http://ua.archive.ubuntu.com/ubuntu xenial/main amd64 manpages-dev all 4.04-2 [2.048 kB]
Получено 2.048 kB за 30с (66,5 kB/с)
(Чтение базы данных ... на данный момент установлено 384084 файла и каталога.)
Подготовка к распаковке ./manpages-dev_4.04-2_all.deb ...
Распаковывается manpages-dev (4.04-2) на замену (4.02-0ubuntu1) ...
Обрабатываются триггеры для man-db (2.7.5-1) ...
Настраивается пакет manpages-dev (4.04-2) ...
sergii@sergii-VirtualBox:~$
```

подключается к телефону
МАЙ 27, 2017



Настройка Ubuntu 16.04 после установки

МАР 17, 2016

РЕКЛАМА

ОПРОС

Какая тема лучше для Ubuntu

Numix (27%, 39 Голосов)

Arc (17%, 24 Голосов)

Adwaita (14%, 20 Голосов)

StylishDark (7%, 10 Голосов)

Adapta (7%, 10 Голосов)

Matcha (4%, 6 Голосов)

Flat Remix (4%, 6 Голосов)

Paper (3%, 5 Голосов)

Materia (3%, 5 Голосов)

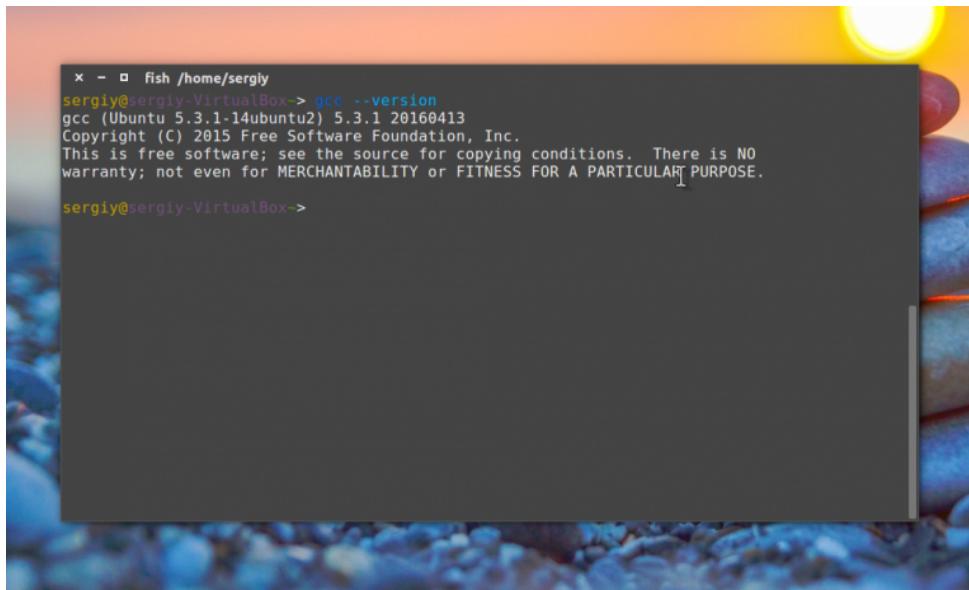
Vimix (3%, 4 Голос

DeLorean Dark (3%, 4 Голосов)

Dorian (2%, 3 Голосов)

Luv (1%, 2 Голосов)

```
$ gcc --version
```



Но перед тем как переходить к самой компиляции программ рассмотрим более подробно составляющие этого процесса.

КАК ВЫПОЛНЯЕТСЯ КОМПИЛЯЦИЯ?

Компиляция программы Linux - это довольно сложный процесс. Все еще сложнее, потому что код программы содержится не в одном файле и даже не во всех файлах ее исходников. Каждая программа использует множество системных библиотек, которые содержат стандартные функции. К тому же один и тот же код должен работать в различных системах, содержащих различные версии библиотек.

На первом этапе, еще до того как начнется непосредственно компиляция, специальный инструмент должен проверить совместима ли ваша система с программой, а также есть ли все необходимые библиотеки. Если чего-либо нет, то будет выдана ошибка и вам придется устранить проблему.

Дальше идет синтаксический анализ и преобразование исходного кода в объектный код, без этого этапа можно было бы и обойтись, но это необходимо, чтобы компилятор мог выполнить различные оптимизации, сделать размер конечной программы меньше, а команды процессора эффективнее.

Затем все объектные файлы собираются в одну программу, связываются с системными библиотеками. После завершения этого

Vertex (1%, 2 Голосов)

Arrangin (1%, 2 Голосов)

Super Flat Remix (1%, 1 Голосов)

Vivacious (0%, 0 Голосов)

Ambiance Dark Blue (0%, 0 Голосов)

Autumn Kerala (0%, 0 Голосов)

Sable Theme (0%, 0 Голосов)

Pop (0%, 0 Голосов)

Всего проголосовавших:
145

ПОДПИСКА НА РАССЫЛКУ

Подписывайтесь на обновления и получайте уведомления о новых интересных статьях по почте.

Введите ваш Email:

☐ I consent to my submitted data being collected via this form*

ПОДПИСАТЬСЯ!

Никакой рекламы или спама!

МЫ В КОНТАКТЕ

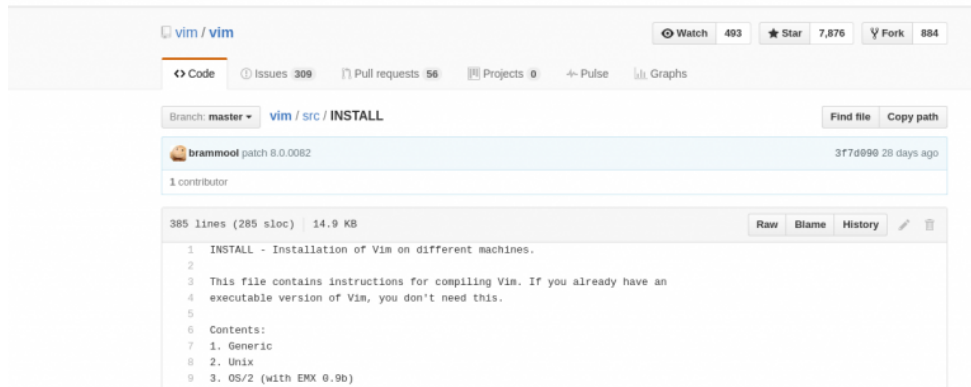
этапа программу остается только установить в файловую систему и
Главная Новости Категории ▾ Рекомендуем ▾ Книги
все. Вот такие основные фазы компиляции программы, а теперь
перейдем ближе к практике.

КОМПИЛЯЦИЯ ПРОГРАММ LINUX

Первое что нам понадобится - это исходники самой программы. В этом примере мы будем собирать самую последнюю версию vim. Это вполне нейтральная программа, достаточно простая и нужная всем, поэтому она отлично подойдет для примера.

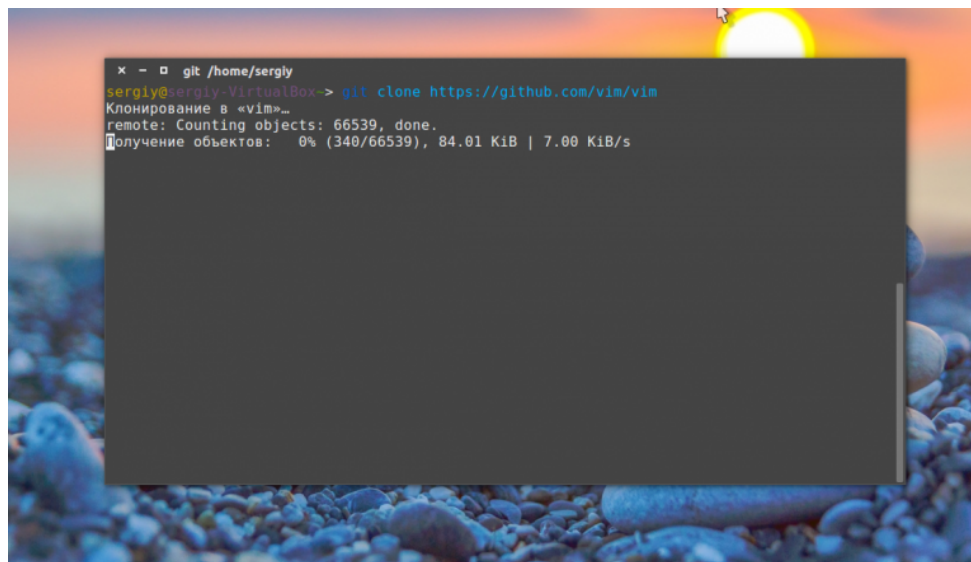
ПОЛУЧЕНИЕ ИСХОДНИКОВ

Первое что нам понадобится, это исходные коды программы, которые можно взять на GitHub. Вы можете найти исходники для большинства программ Linux на GitHub. Кроме того, там же есть инструкции по сборке:



Давайте загрузим сами исходники нашей программы с помощью утилиты git:

```
$ git clone https://github.com/vim/vim
```



Опросы

Losst ▾

komYounity | Linux Ubuntu

Мы создаём комьюнити, комьюнит создаёт Linux.

10 814 подписчиков



Подписаться на новост

ОБНАРУЖИЛИ ОПЕЧАТКУ?

Сообщите нам об этом, выделите текст с ошибкой и нажмите **Ctrl+Enter**, будем очень признательны!

МЕТА

- [Регистрация](#)
- [Войти](#)
- [Entries RSS](#)
- [Comments RSS](#)

ОСТАВАЙТЕСЬ С НАМИ В СОЦИАЛЬНЫХ СЕТЯХ:



его в нужную папку, но так будет удобнее. Утилита создаст папку с именем программы, нам нужно сделать ее рабочей:

```
$ cd vim
```

НАСТРОЙКА CONFIGURE

Дальше нам нужно запустить скрипт, который проверит нашу программу на совместимость с системой и настроит параметры компиляции. Он называется configure и поставляется разработчиками программы вместе с исходниками. Весь процесс компиляции описан в файле Makefile, его будет создавать эта утилита.

Если configure нет в папке с исходниками, вы можете попытаться выполнить такие скрипты чтобы его создать:

```
$ ./bootstrap
$ ./autogen.sh
```

Также для создания этого скрипта можно воспользоваться утилитой automake:

```
$ aclocal
$ autoheader
$ automake --gnu --add-missing --copy --foreign
$ autoconf -f -Wall
```

Утилита automake и другие из ее набора генерируют необходимые файлы на основе файла Makefile.am. Этот файл обязательно есть в большинстве проектов.

После того как вы получили configure мы можем переходить к настройке. Одним из огромных плюсов ручной сборки программ есть то, что вы можете сами выбрать с какими опциями собирать программу, где она будет размещена и какие дополнительные возможности стоит включить. Все это настраивается с помощью

МЫ В TWITTER

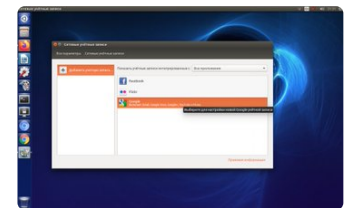
ТВИТЫ от @Losst_official ⓘ



Losst

@Losst_official

#Linux #Списки Лучшие клиенты Google Drive Linux losst.ru/luchshie-klien...



4...



Losst

@Losst_official

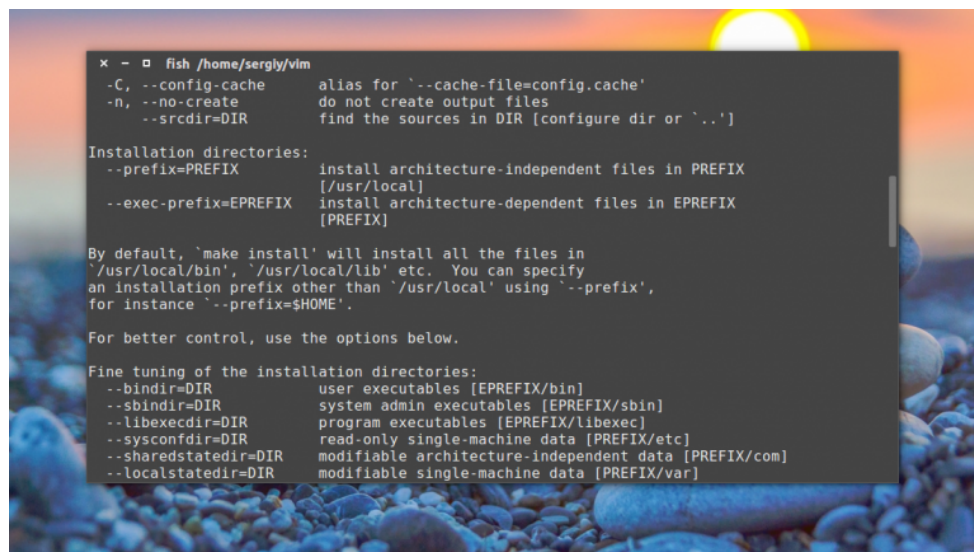
#Linux #Инструкции Минимальная установка Fedora losst.ru/minimalnaya-us...

[Встроить](#)

[Показать в Твиттере](#)




```
$ ./configure --help
```

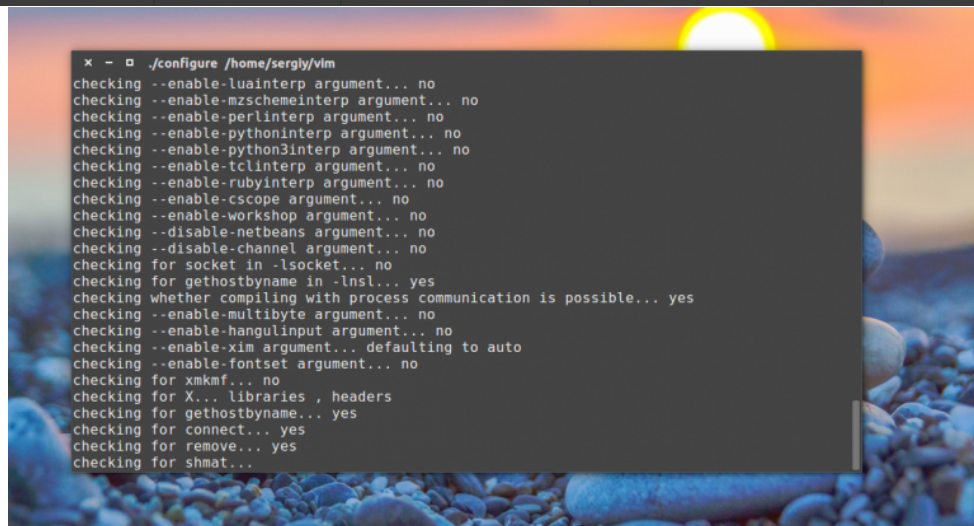


Рассмотрим наиболее часто используемые, стандартные для всех программ опции:

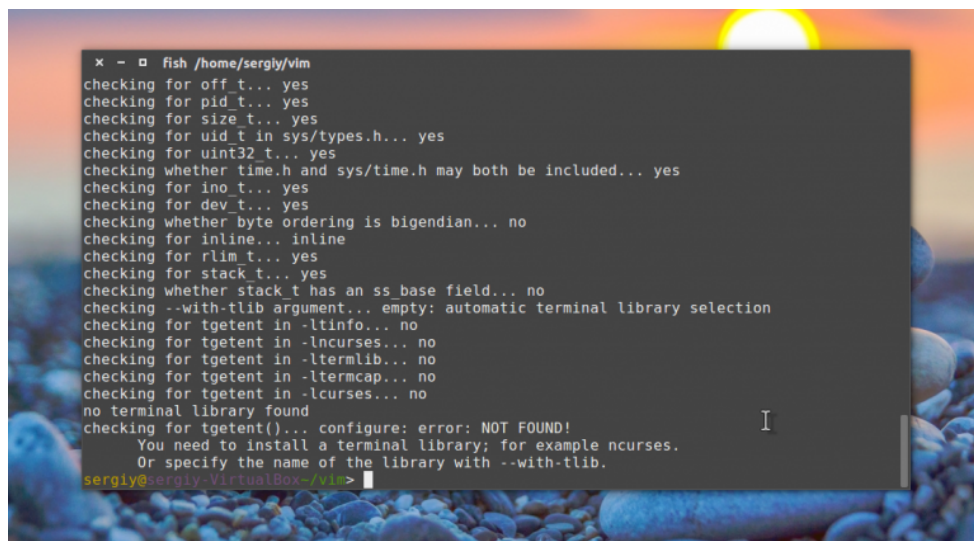
- **--prefix=PREFIX** - папка для установки программы, вместо /, например, может быть /usr/local/, тогда все файлы будут распространены не по основной файловой системе, а в /usr/local/;
- **--bindir=DIR** - папка для размещения исполняемых файлов, должна находиться в PREFIX;
- **--libdir=DIR** - папка для размещения и поиска библиотек по умолчанию, тоже в PREFIX;
- **--includedir=DIR** - папка для размещения map страниц;
- **--disable-возможность** - отключить указанную возможность;
- **--enable-возможность** - включить возможность;
- **--with-библиотека** - подобно enable активирует указанную библиотеку или заголовочный файл;
- **--without-библиотека** - подобное disable отключает использование библиотеки.

Вы можете выполнить configure без опций, чтобы использовать значения по умолчанию, но также можете вручную указать нужные пути. В нашем случае ./configure есть, и мы можем его использовать:





Во время настройки утилита будет проверять, есть ли все необходимые библиотеки в системе, и если нет, вам придется их установить или отключить эту функцию, если это возможно. Например, может возникнуть такая ошибка: **no terminal library found checking for tgetent()... configure: error: NOT FOUND!**



В таком случае нам необходимо установить требуемую библиотеку. Например, программа предлагает ncurses, поэтому ставим:

```
$ sudo apt install libncurses-dev
```

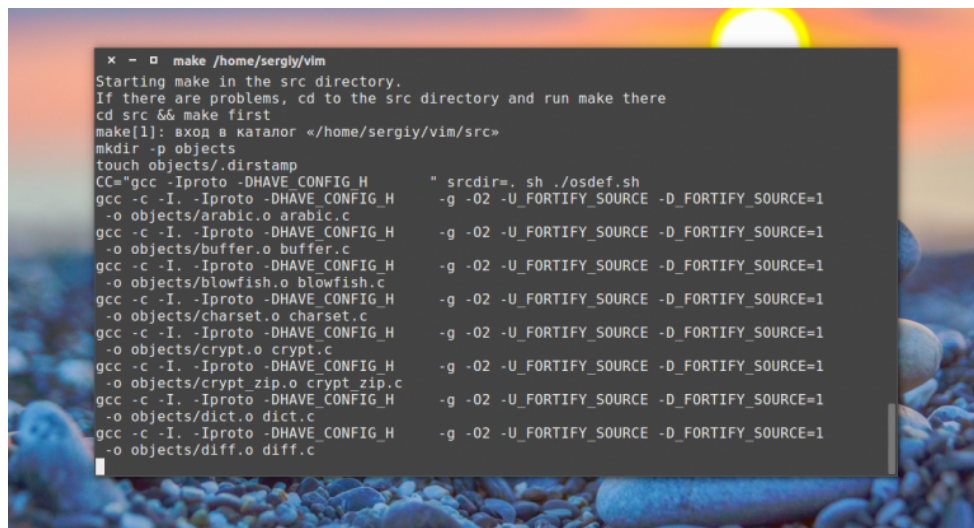
Приставка lib всегда добавляется перед библиотеками, а -dev - означает, что нам нужна библиотека со всеми заголовочными файлами. После удовлетворения всех зависимостей настройка пройдет успешно.

СБОРКА ПРОГРАММЫ



этапе выполняется непосредственно преобразование исходного кода в машинный. Утилита make на основе Makefile делает все необходимые действия:

```
$ make
```



Дальше осталось установить саму программу, если вы использовали опцию prefix, чтобы не устанавливать программу в основную файловую систему, то можно применить стандартную опцию make:

```
$ make install
```

После этого программа будет установлена в указанную вами папку, и вы сможете ее использовать. Но более правильный путь - создавать пакет для установки программы, это делается с помощью утилиты checkinstall, она позволяет создавать как deb, так и rpm пакеты, поэтому может использоваться не только в Ubuntu. Вместо make install выполните:

```
$ checkinstall
```

Затем просто установите получившийся пакет с помощью dpkg:

```
$ sudo dpkg install vim.deb
```

После этого сборка программы полностью завершена и



Если вы устанавливали программу с помощью `make install`, то удалить ее можно выполнив в той же папке обратную команду:

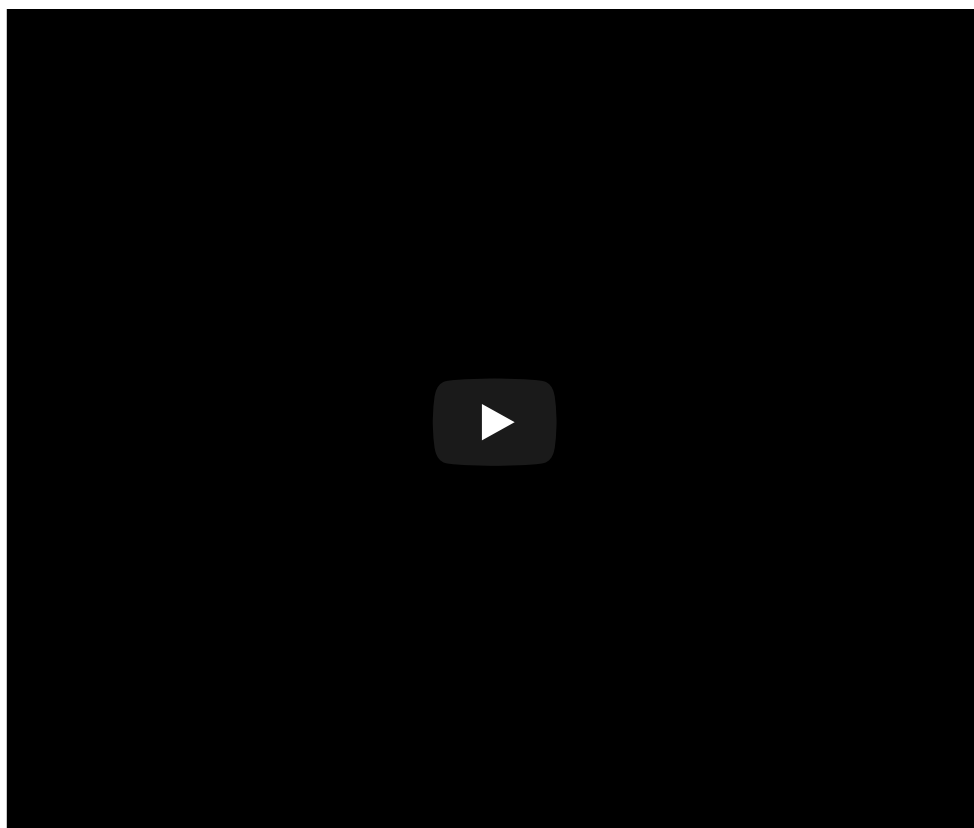
```
$ sudo make uninstall
```

Команда удалит все файлы, которые были скопированы в файловую систему.

Выводы

В этой статье мы рассмотрели, как выполняется компиляция программы Linux. Этот процесс может быть сложным для новичков, но в целом, все возможно, если потратить на решение задачи несколько часов. Если у вас остались вопросы, спрашивайте в комментариях!

На завершение видео о том, что такое компилятор и интерпретатор:



Похожие записи:

Нет похожих записей

Оцените Статью:

★★★★★ (11 оценок, среднее: **4,64** из 5)

ОБ АВТОРЕ



ADMIN

Основатель и администратор сайта losst.ru, увлекаюсь открытым программным обеспечением и операционной системой Linux. В качестве основной ОС сейчас использую Ubuntu. Кроме Linux интересуюсь всем, что связано с информационными технологиями и современной наукой.

Ваше имя тоже может быть здесь. Начните писать статьи для Losst. Это просто! Смотрите подробнее как начать писать статьи -

[Пишите для нас](#)

8 КОММЕНТАРИЕВ



VITAL ДЕКАБРЬ 10, 2016

[↪ Ответить](#)

Люблю Ubuntu - не люблю убунту юзеров : "AAAAA не работаит!Афтар дИбил!!"
В дженту вас сослать на повышение квалификации.



UNDERFUND ДЕКАБРЬ 10, 2016

[↪ Ответить](#)

Программируйте на Phyton, интерпретируемый и к тому же OpenSource, кроссплатформенный в отличии от этих C/C++. Написал код/скрипт и он тут же работает, не требуется секса с компиляцией, так то.



CAMINO ДЕКАБРЬ 11, 2016

[↪ Ответить](#)



**YGOR** АПРЕЛЬ 6, 2018[↪ Ответить](#)

Ни один из способов компиляции не помог. Нет там никаких файлов `configure`, `bootstrap`, `autogen.sh`, не работают команды и далее которые выдаёт `'configure.ac' or 'configure.in' is required` в папке исходников вообще никаких запускаемых файлов нет. Есть файлы с расширением `stake` и ещё куча разных других непонятных помимо самого кода и заголовочных. Инструкций тоже никаких нет.

**GARYNYCH** ИЮЛЬ 23, 2018[↪ Ответить](#)

обычно авторы программ у себя на гитхабе (там где расположены исходники) или в файле `README.md` описывают способ компиляции, установки и т.п.

**ЮРИЙ ДЕНИСОВ** ИЮЛЬ 25, 2018[↪ Ответить](#)

У меня получается собирать проект утилитой `checkinstall` только под `sudo`. Собираю `freeciv` версии 2.6.0, т.к. готовая сборка этой программы есть только под винду, а на убунте устаревшая версия 2.5.x. Пакет собирается, но не устанавливается, но это уже другая история))

**TAXWO** АПРЕЛЬ 19, 2019[↪ Ответить](#)

Всё получилось, спасибо. Только в конце мой `Debian 9` не захотел исполнять `sudo dpkg install vim.deb` Но



`sudo dpkg -i vim.deb` взял в обработку без проблем, да и еще, не успел даже проверить название собранного пакета в папке `vim`. У меня он такой - `vim_20190419-1_amd64.deb`. Просто новички так и будут писать, как у вас в статье имя пакета.



SERGEY75

АВГУСТ 3, 2019

Ответить

А как на Арче?



ОТВЕТИТЬ

Текст комментария*

Имя*

Email*

Веб-сайт

ДОБАВИТЬ КОММЕНТАРИЙ

